Matthew Spahl, Emerson Kiefer, Kevin Martell Luya, Owen Tibby

CS 532 Final Project Instructions for running code

(also providing as a txt file titled “README”, but this is a bit longer and allows for pictures)

This program reads in a dataset of used car pricing information, trains a random forest on that dataset, and then predicts the RV_percent (retained value percent) for each sample in the test data.

Code is contained in the file Pyspark_forest.py

Download the dataset below and put it in the same directory as Pyspark_forest.py. The data file should be named: cleaned_data_CS532.csv

Link to data set that has been further cleaned (the one to use with the program):

https://drive.google.com/file/d/1X4a0lvQ9_nXYGUkL6uiyy5Zvu2wYlhEP/view?usp=sharing

Original Dataset from Kaggle (for reference but not the one to use with the program):

https://www.kaggle.com/datasets/tunguz/used-car-auction-prices/data

References consulted to learn how to use a random forest classifier in PySpark and measure time elapsed:

https://towardsdatascience.com/a-guide-to-exploit-random-forest-classifier-in-pyspark-46d6999cb5db

https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.regression.RandomForestRegressor.html

https://www.machinelearningplus.com/pyspark/pyspark-onehot-encoding/

https://spark.apache.org/docs/latest/api/python/reference/api/pyspark.ml.evaluation.RegressionEvaluator.html

https://stackoverflow.com/questions/7370801/how-do-i-measure-elapsed-time-in-python

Instructions for running code: in an Ubuntu VM with python and pip installed, install numpy and pandas.

To run, inside the directory with the Pyspark_forest.py file, use command: python3 Pyspark_forest.py

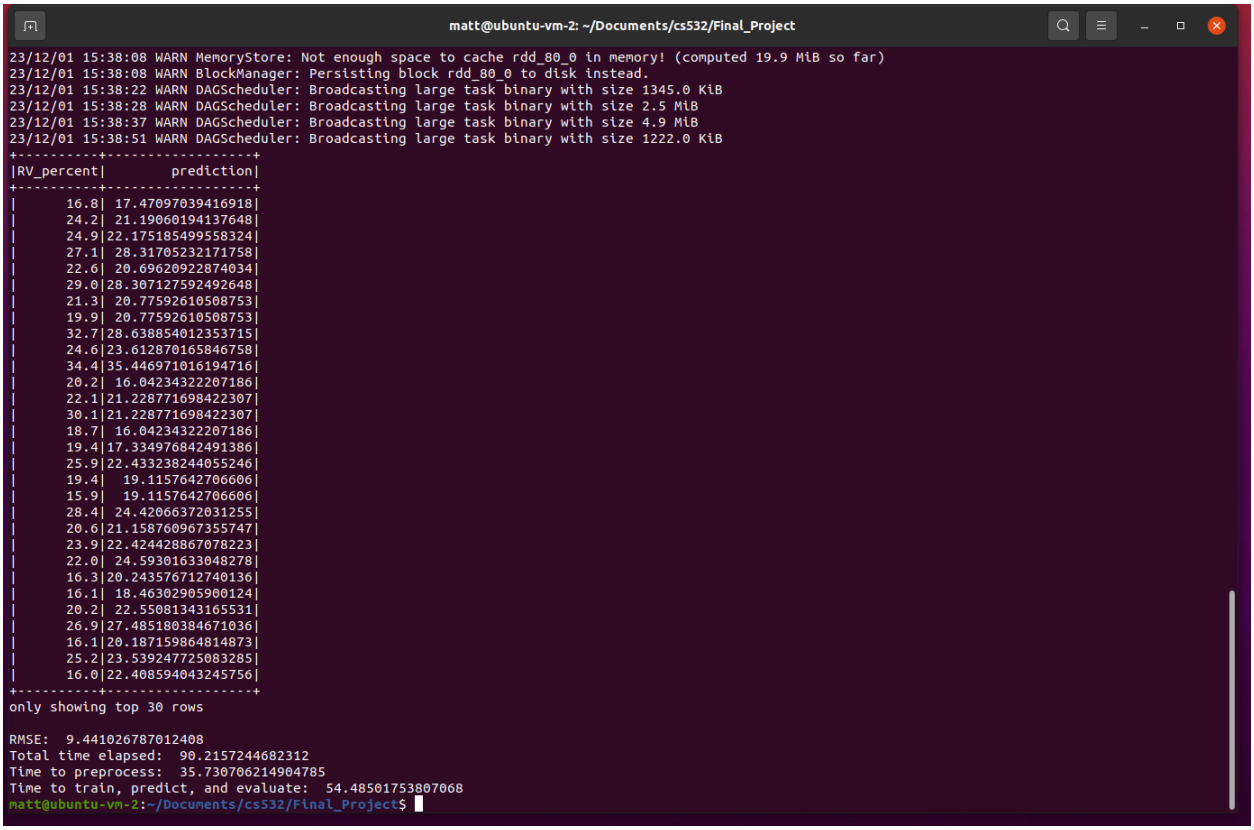
By default, the program will train/fit the random forest model using 50 trees. To change this, either change the value of treeCount on line 110 of the program or provide a command line argument. For example, to run with 100 trees, use the command:

python3 Pyspark_forest.py 100

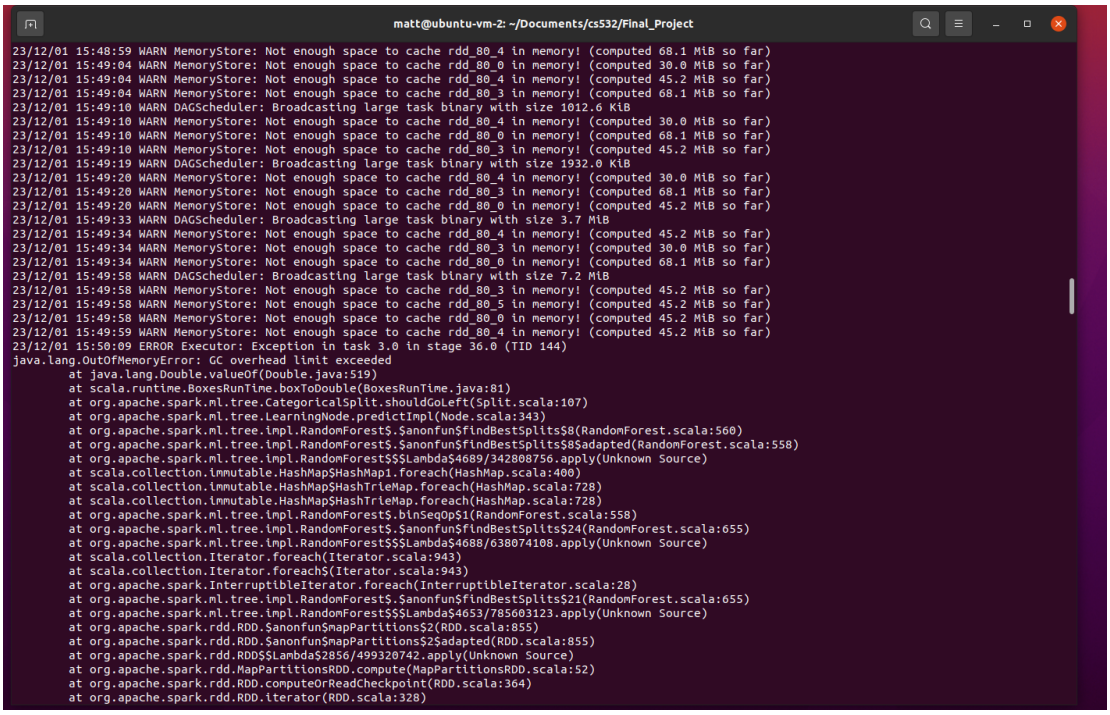
The program will process the data and train a random forest regressor model and make predictions, then output the total runtime of the program, the data processing time, and the time for train/fit the random forest, make predictions, and evaluate the model.

Also, in our analysis for our presentation, referenced spark.apache.org/docs/2.2.0/mllib-ensembles.html to confirm that Pyspark Random Forest is able to train trees in parallel.

Example output from 6GB RAM and 6 cores, 200 trees:



As explained in our presentation, when using 2 or 4 cores, works for up to 400 trees, but with 6 cores, worked for up to 200 trees, then get the following out of memory error if we try to use 300 trees (we document this in our presentation, this is due to PySpark using parallelism with more cores and trying to train multiple trees at once):



For our experiments, we ran the program changing VM settings – we tested with 2GB, 4GB, and 6GB RAM, and 2 cores, 4 cores, and 6 cores. Tested 10-200 trees for all settings, and sometimes up to 400 for 2 cores or 4 cores. See presentation for full results and description of tests.